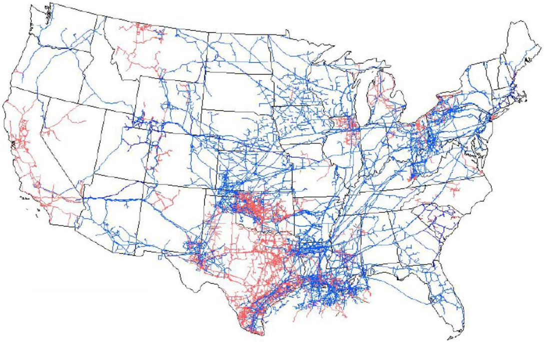
Stochastic Optimization of Gas Networks

Nai-Yuan Chiang[1] Victor M Zavala[1]

[1]Mathematics and Computer Science, Argonne National Laboratory

April 01 2014

Motivation



U.S. Gas Transmission System

- How to exploit mutiple problem embedded structure?
- Tools:

  Structure Exploiting Parallel Interior-Point Solver for NLP

Outline

Interior Point Methods (IPM)

Nonlinear Program

$$\min \mathbf{f}(x) \qquad \text{s.t.} \quad \begin{aligned} \mathbf{c}(x) &= 0 \\ x &\geq 0 \end{aligned} \qquad \text{(NLP)}$$

KKT Conditions

$$\begin{aligned} \nabla \mathbf{f}(x) - \nabla \mathbf{c}(x)\lambda - s &= 0 \\ \nabla \mathbf{c}^\top x &= 0 \\ XSe &= 0 \\ x, s &\geq 0 \end{aligned} \qquad \text{(KKT)}$$

$X = \mathrm{diag}(x), S = \mathrm{diag}(s)$

Interior Point Methods (IPM)

Barrier Problem

$$\min \mathbf{f}(x) - \mu \sum \ln x_i \quad \text{s.t.} \quad \begin{aligned} \mathbf{c}(x) &= 0 \\ x &\geq 0 \end{aligned} \qquad (\text{NLP}_\mu)$$

KKT Conditions

$$\begin{aligned} \nabla \mathbf{f}(x) - \nabla \mathbf{c}(x)\lambda - s &= 0 \\ \nabla \mathbf{c}^\top x &= 0 \\ XSe &= \mu e \\ x, s &\geq 0 \end{aligned} \qquad (\text{KKT}_\mu)$$

$X = \mathrm{diag}(x), S = \mathrm{diag}(s)$

- Introduce logarithmic barriers for $x \geq 0$
- For $\mu \to 0$ solution of (NLP_μ) converges to solution of (NLP)
- System (KKT_μ) can be solved by Newton's Method

Newton-Step in IPM

Newton-Step: Full System

$$\begin{bmatrix} H & \mathcal{A}^\top & -I \\ \mathcal{A} & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} \nabla f + \mathcal{A}^\top y - z \\ c(x) \\ XZe - \mu e \end{bmatrix}$$

where $\mathcal{A}$ is the constraint Jacobian, and H is the Hessian of the Lagrangian function. $\Theta = X^{-1}S, \quad X = diag(x), \quad S = diag(s)$.

Newton-Step: Augmented System(IPM) $\Phi d = b$

$$\begin{bmatrix} H + \Theta & \mathcal{A}^\top \\ \mathcal{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = - \begin{bmatrix} \nabla \phi_\mu + \mathcal{A}^\top y \\ c(x) \end{bmatrix}$$

- Augmented system is sparse and symmetric.
- The structure of Newton's system does not change between IPM iterations.

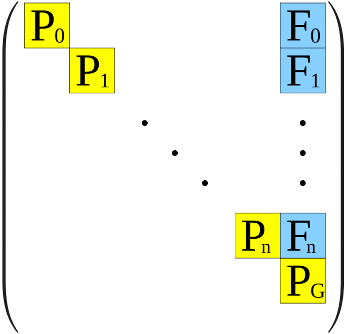
Parallel Linear Algebra for IPM

Newton-Step: Augmented System(IPM)

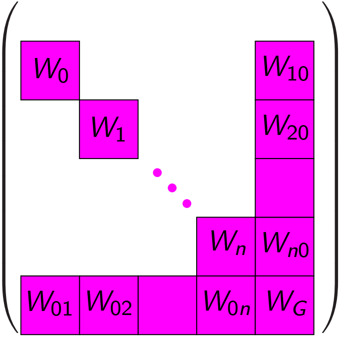
$$\begin{bmatrix} W & \mathcal{A}^\top \\ \mathcal{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \end{bmatrix}, W = H + \Theta$$

For a structured problem, if:

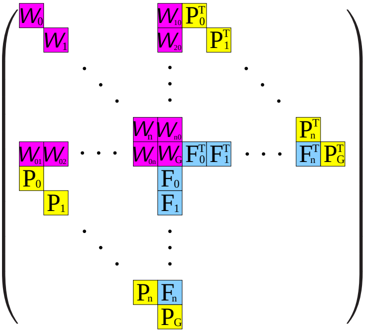
Matrix $\mathcal{A}$



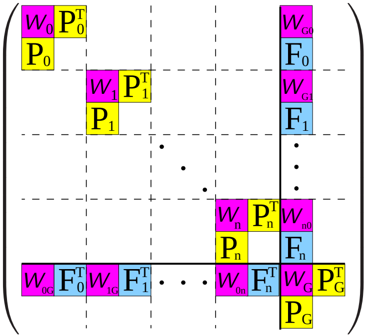
Matrix W



Structures of $\mathcal{A}$, W and Φ:

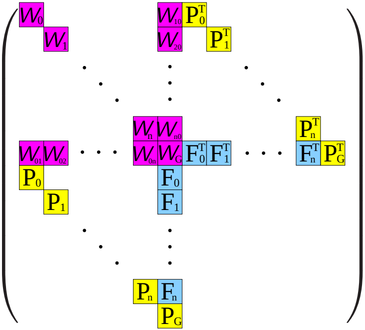


$$\begin{pmatrix} Q & \mathcal{A}^\top \\ \mathcal{A} & 0 \end{pmatrix}$$

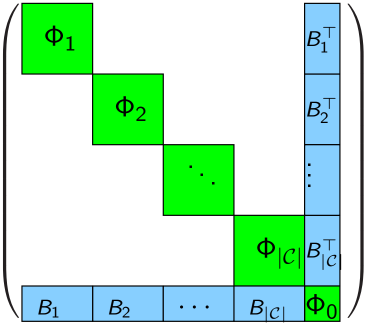


$$P \begin{pmatrix} Q & \mathcal{A}^\top \\ \mathcal{A} & 0 \end{pmatrix} P^{-1}$$

Structures of $\mathcal{A}$, W and Φ:



$$\begin{pmatrix} Q & \mathcal{A}^\top \\ \mathcal{A} & 0 \end{pmatrix}$$



$$P \begin{pmatrix} Q & \mathcal{A}^\top \\ \mathcal{A} & 0 \end{pmatrix} P^{-1}$$

Bordered block-diagonal structure in Augmented System!

Exploiting Structure in IPM

Block-Factorization of Augmented System Matrix

$$\underbrace{\begin{pmatrix} \Phi_1 & & & B_1^\top \\ & \ddots & & \vdots \\ & & \Phi_n & B_n^\top \\ B_1 & \cdots & B_n & \Phi_0 \end{pmatrix}}_{\Phi} \underbrace{\begin{pmatrix} x_1 \\ \vdots \\ x_n \\ x_0 \end{pmatrix}}_{d} = \underbrace{\begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \\ \mathbf{b}_0 \end{pmatrix}}_{\mathbf{b}}$$

Solution of Block-system by Schur-complement

The solution to $\Phi x = \mathbf{b}$ is

$$\begin{aligned} x_0 &= C^{-1}\mathbf{b}_0, \quad \mathbf{b}_0 = b_0 - \textstyle\sum_i B_i \Phi_i^{-1} \mathbf{b}_i \\ x_i &= \Phi_i^{-1}(\mathbf{b}_i - B_i^\top x_0), \qquad i = 1, \ldots, n \end{aligned}$$

where C is the *Schur-complement*

$$C = \Phi_0 - \sum_{i=1}^{n} B_i \Phi_i^{-1} B_i^\top$$

$\Rightarrow$ only need to factor Φ_i, not Φ

Paraller Linear Algebra for the Structured Problem

Parallel IPM Implementation: For LP and QP

- OOPS: Jacek Gondzio and Andreas Grothey: Exploiting structure in parallel implementation of interior point methods for optimization.
- PIPS: Cosmin G. Petra and Mihai Anitescu: A preconditioning technique for Schur complement systems arising in stochastic optimization.

Now, we have PIPS-NLP for NLP!

PIPS-NLP

PIPS-NLP

PIPS-NLP is parallel nonliear IPM solver, based on PIPS for LP/QP.

Structure comes from (but not limited to)

- Stochastic Programming (scenarios)
- Problem Characteristics (PDE constraints)
- Network (partitions)
- Nested structure

Easy access:

- AMPL-interface
- Pyomo-interface

Global Convergence of the Algorithm

- NLP needs more work to ensure global convergence: filter technique (IPOPT[1]), which requires inertia (number of positive and negative eigenvalue).

Structured Problem: Inertia Detection may be hard.

- $\mathit{Inertia}(\Phi) = \mathit{Inertia}(C) + \sum_i \mathit{Inertia}(\Phi_i)$
  Not Clear How to Do it (Central vs. Block-Based)
- Schur Complement is Large or Indefinite

[1] Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Math. Program.* 106.1, Ser. A (2006), pp. 25–57. ISSN: 0025-5610.

Global Convergence of the Algorithm

Cases Where Inertia Detection is Difficult (If Not Impossible):

- Full System or Individual Blocks are Solved Using Iterative Schemes, e.g, multi-grid
- Numerically unstable on large-scale prob. (Some linear system solvers can give us inertia infomation, e.g. MA57, but its pivot tolerence plays a very important role.)
- Reduced Solver is Applied.
- Nested Structure.

Line Search with Filter

Inertia Correction: Rigorous Detection

- Check if matrix $\Phi = \begin{bmatrix} W & \mathcal{A}^\top \\ \mathcal{A} & 0 \end{bmatrix}$ has correct inertia. (MA57 or Pardiso)
- Increase the regularization term δ in $\Phi_\delta = \begin{bmatrix} W + \delta I & \mathcal{A}^\top \\ \mathcal{A} & 0 \end{bmatrix}$, until its inertia is correct.
- Solve system $\Phi_\delta d = b$.

dWd test: Relaxed Detection

- Check if $d^\top W d$ has sufficient curvature for global convergence:
  $d^\top W d \geq \theta d^\top d$, where θ is a constant decreasing by μ.
- Increase regularization 'only' for the iteration whose $d^\top (W + \delta I) d < \theta d^\top d$.

Relaxed Curvature test

Pros and Cons of Relaxed Detection

- Pros:
  Can accept decent direction d even if current inertia is wrong. Global convergence is also guaranteeded. Re-factorization is expensive! $>90\%$ time of each Iter!
- Cons:
  Cannot guarantee second order optimality condition, but neither does inertia correction.

In practice?

Relaxed Curvature test

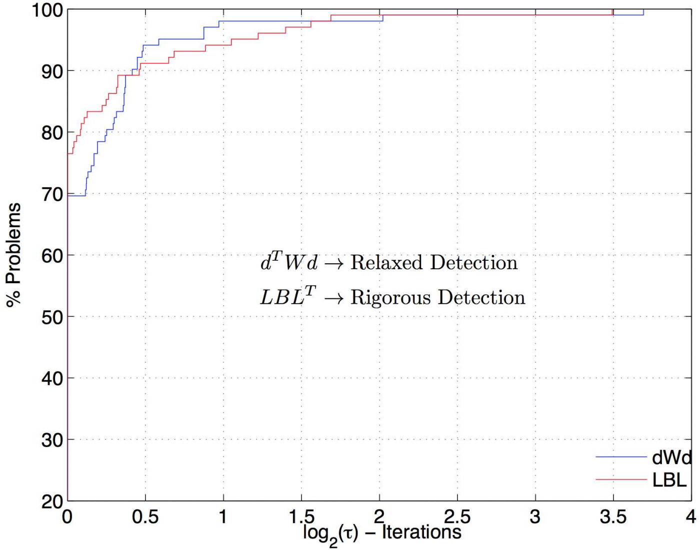
Pros and Cons of Relaxed Detection

- Pros:
  Can accept decent direction *d* even if current inertia is wrong. Global convergence is also guaranteeded. Re-factorization is expensive! $> 90\%$ time of each Iter!
- Cons:
  Cannot guarantee second order optimality condition, but neither does inertia correction.

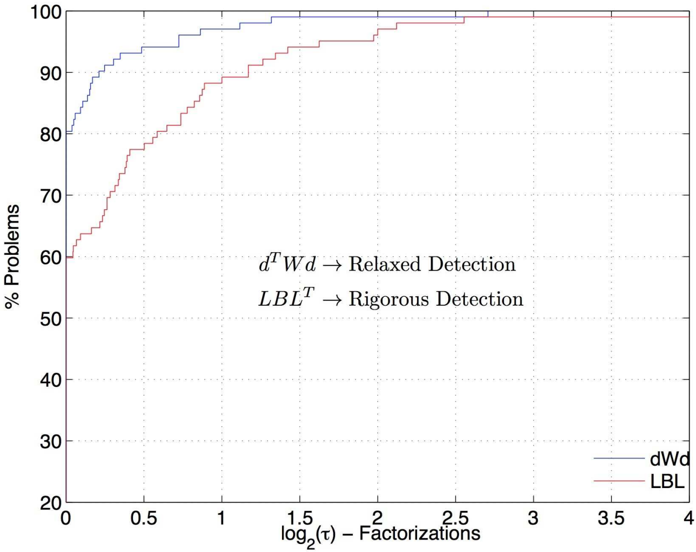
In practice?

- CUTEr test problems.
- Energy application.

CUTEr Experiments : (iteration)



CUTEr Experiments : (factorization)

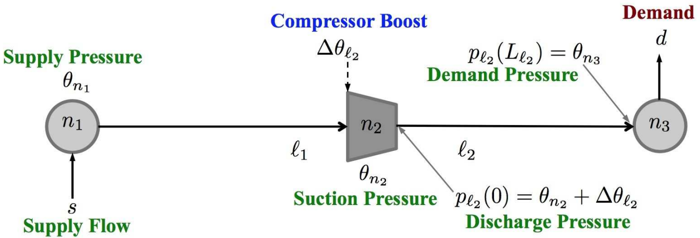


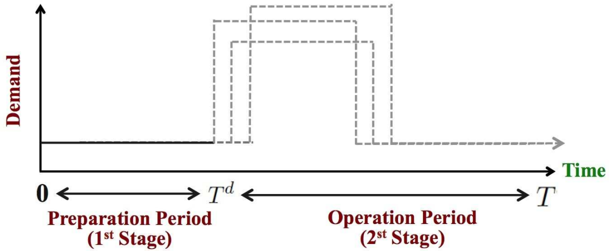
Application.:

Problem	Domain	Algorithm	Objective	Iter	Fact
building_det	Buildings	$d^\top Wd$	1.74×10^3	127	161
		$LDL^\top$	1.74×10^3	180	341
building_stoch_A	Buildings	$d^\top Wd$	1.89×10^3	167	181
		$LDL^\top$	1.89×10^3	170	270
building_stoch_B	Buildings	$d^\top Wd$	1.95×10^3	319	417
		$LDL^\top$	1.95×10^3	170	270
IEEE_162cart	Power Grid	$d^\top Wd$	1.64×10^0	23	23
		$LDL^\top$	1.64×10^0	104	330
stochPDEgas_A	Gas Network	$d^\top Wd$	1.73×10^2	35	35
		$LDL^\top$	1.73×10^2	34	35

Stochastic Gas Networks: Stochastic Structure + PDE Structure

Line-Pack (Storage) Management





PDE Gas Model

Inlet & Outlet Pipe Flows — **Supply & Demand Flows**

Network

$$0 = \sum_{\ell \in \mathcal{L}_n^{in}} f_{\ell,t}^{in}(\omega) - \sum_{\ell \in \mathcal{L}_n^{out}} f_{\ell,t}^{out}(\omega) + \sum_{i \in \mathcal{S}_n} s_{i,t}(\omega) - \sum_{j \in \mathcal{D}_n} d_{j,t}(\omega),\ n \in \mathcal{N}, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

Conservation & Momentum

$$\frac{p_{\ell,t+1,k}(\omega) - p_{\ell,t,k}(\omega)}{\Delta\tau} = -c_{1,\ell}\frac{f_{\ell,t+1,k+1}(\omega) - f_{\ell,t+1,k}(\omega)}{\Delta x_\ell},\ \ell \in \mathcal{L}, t \in \bar{\mathcal{T}}^-, k \in \bar{\mathcal{X}}^-, \omega \in \Omega$$

$$\frac{f_{\ell,t+1,k}(\omega) - f_{\ell,t,k}(\omega)}{\Delta\tau} = -c_{2,\ell}\frac{p_{\ell,t+1,k+1}(\omega) - p_{\ell,t+1,k}(\omega)}{\Delta x_\ell} - c_{3,\ell}\frac{f_{\ell,t+1,k}(\omega)|f_{\ell,t+1,k}(\omega)|}{p_{\ell,t+1,k}(\omega)},\ \ell \in \mathcal{L}, t \in \bar{\mathcal{T}}^-, k \in \bar{\mathcal{X}}^-, \omega \in \Omega$$

Boundary Conditions

Pipe Flows

$$f_{\ell,t,N_x}(\omega) = f_{\ell,t}^{out}(\omega),\ \ell \in \mathcal{L}, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

$$f_{\ell,t,1}(\omega) = f_{\ell,t}^{in}(\omega),\ \ell \in \mathcal{L}, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

$$p_{\ell,t,N_x}(\omega) = \theta_{rec(\ell),t}(\omega),\ \ell \in \mathcal{L}, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

Pipe Pressures

$$p_{\ell,t,1}(\omega) = \theta_{snd(\ell),t}(\omega),\ \ell \in \mathcal{L}_p, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

$$p_{\ell,t,1}(\omega) = \theta_{snd(\ell),t}(\omega) + \Delta\theta_{\ell,t}(\omega),\ \ell \in \mathcal{L}_a, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

Compressor Power

$$P_{\ell,t}(\omega) = c_4\, f_{\ell,t}^{in}(\omega)\left(\left(\frac{\theta_{snd(\ell),t}(\omega) + \Delta\theta_{\ell,t}(\omega)}{\theta_{snd(\ell),t}(\omega)}\right)^\beta - 1\right),\ \ell \in \mathcal{L}_a, t \in \bar{\mathcal{T}}, \omega \in \Omega.$$

PDE Gas Model: Objective

Define Per Scenario Cost

$$\varphi(\omega) = \underbrace{\sum_{t\in\bar{\mathcal{T}}}\sum_{\ell\in\mathcal{L}_a} c_{e,t}P_{\ell,t}(\omega)\Delta\tau}_{\text{Power}} + \underbrace{\sum_{t\in\bar{\mathcal{T}}}\sum_{i\in\mathcal{D}} c_d(d_{j,t}(\omega)-\bar{d}_{j,t}(\omega))^2\Delta\tau}_{\text{Demand Tracking}}$$
$$+ \underbrace{\sum_{k\in\bar{\mathcal{X}}}\sum_{\ell\in\mathcal{L}} c_T(p_{\ell,T,k}(\omega)-p_{\ell,1,k}(\omega))^2\Delta x_\ell + \sum_{k\in\bar{\mathcal{X}}}\sum_{\ell\in\mathcal{L}} c_T(f_{\ell,T,k}(\omega)-f_{\ell,1,k}(\omega))^2\Delta x_\ell}_{\text{Terminal Constraints}}, \quad \omega\in\Omega.$$

Objective Function

$$\Psi = (1-\xi)\mathbb{E}\left[\varphi(\omega)\right] + \xi\mathrm{CVaR}\left[\varphi(\omega)\right].$$

where,

$$\mathrm{CVaR}\left[\varphi(\omega)\right] = \min_{\nu}\left[\nu + \frac{1}{1-\sigma}\mathbb{E}\left[\varphi(\omega)-\nu\right]_+\right]$$

- **Soft Constraints to Enforce Demand Flows (Improves Flexibility)**
- **Terminal Constraints Critical (System Required to Return to Initial State)**

PDE Gas Model: System Constraint

$$\theta_{sup(i),t}(\omega) = \bar{\theta}_i^{sup}, \; i \in \mathcal{S}, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

Supply Pressures

$$\Delta\theta_{\ell,1}(\omega) = \Delta\theta_\ell^0, \; \ell \in \mathcal{L}_a, \omega \in \Omega$$

$$0 = -c_{1,\ell}\frac{f_{\ell,1,k+1}(\omega) - f_{\ell,1,k}(\omega)}{\Delta x_\ell}, \; \ell \in \mathcal{L}, x \in \bar{\mathcal{X}}^-, \omega \in \Omega$$

$$0 = -c_{2,\ell}\frac{p_{\ell,1,k+1}(\omega) - p_{\ell,1,k}(\omega)}{\Delta x_\ell} - c_{3,\ell}\frac{f_{\ell,1,k}(\omega)|f_{\ell,1,k}(\omega)|}{p_{\ell,1,k}(\omega)}, \; \ell \in \mathcal{L}, x \in \bar{\mathcal{X}}^-, \omega \in \Omega.$$

Initial Conditions

$$P_\ell^L \le P_{\ell,t}(\omega) \le P_\ell^U, \; \ell \in \mathcal{L}_a, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

$$\theta_\ell^{suc,L} \le \theta_{snd(\ell),t}(\omega) \le \theta_\ell^{suc,U}, \; \ell \in \mathcal{L}_a, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

$$\theta_\ell^{dis,L} \le \theta_{snd(\ell),t}(\omega) + \Delta\theta_{snd(\ell),t}(\omega) \le \theta_\ell^{dis,U}, \; \ell \in \mathcal{L}_a, t \in \bar{\mathcal{T}}, \omega \in \Omega$$

$$\theta_j^L \le \theta_{dem(j),t}(\omega) \le \theta_j^{dem,U}, \; j \in \mathcal{D}, t \in \bar{\mathcal{T}}, \omega \in \Omega.$$

Initial Conditions

$$\Delta\theta_{\ell,t}(\omega) = \mathbb{E}\left[\Delta\theta_{\ell,t}(\omega)\right], \; \ell \in \mathcal{L}_a, t \in \{1..T^d\}, \omega \in \Omega \setminus \{1\}$$

Non-Anticipativity

Gas Network Problem: Stochastic Structure

$$\begin{pmatrix} P_0 & F_0 \end{pmatrix}$$

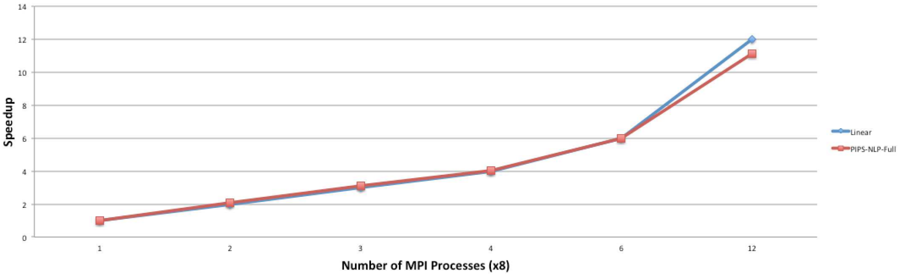
$$\begin{pmatrix} P_0 & & & & F_0 \\ & P_1 & & & F_1 \\ & & \ddots & & \vdots \\ & & & P_{|C|} & F_{|C|} \end{pmatrix}$$

Problem (1 Scenario) Problem with $|C|+1$ Scenarios

- Control variables are independent to scenarios.

Numerical Results: Scalability

No.Sce	n	Obj	Iters	Time(hh:mm:ss)	MPI Proc.
96	1,930,752	1.39×10^2	42	05:20:01	1
96	1,930,752	1.39×10^2	41	01:05:35	8
96	1,930,752	1.39×10^2	40	00:31:15	16
96	1,930,752	1.39×10^2	42	00:21:02	24
96	1,930,752	1.39×10^2	41	00:16:13	32
96	1,930,752	1.39×10^2	41	00:10:59	48
96	1,930,752	1.39×10^2	41	00:05:53	96



Gas Network Problem: PDE Structure

Newton-Step: "Augmented System(IPM)" for each scenario

$$\Phi_i = \begin{bmatrix} W_i & \mathcal{A}_i^\top \\ \mathcal{A}_i & 0 \end{bmatrix}, \forall i \in S$$

where $\mathcal{A}_i$ contains PDE constraints.

Split $\mathcal{A}_i$ into $[\mathcal{A}_x \ \mathcal{A}_u]$ (ignore 'i'), where u is the contral variables and x is the state variables.

$$\Phi_i \mathbf{d}_i = \mathbf{b}_i \rightarrow \begin{bmatrix} W_{xx} & W_{xu} & \mathcal{A}_x^\top \\ W_{ux} & W_{uu} & \mathcal{A}_u^\top \\ \mathcal{A}_x & \mathcal{A}_u & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_u \\ \mathbf{b}_\lambda \end{bmatrix}$$

$\mathcal{A}_x$ is square and invertible (but not symmetric)

$\rightarrow$ Do computation in the reduced space!

Reduced Space

Solve Problem in Reduced Space!

- Step.1: Solve following equation to get u:

$$\begin{aligned}(W_{uu} - W_{ux}\mathcal{A}_x^{-1}\mathcal{A}_u - \mathcal{A}_u^\top \mathcal{A}_x^{-\top} W_{xu} + \mathcal{A}_u^\top \mathcal{A}_x^{-\top} W_{xx}\mathcal{A}_x^{-1}\mathcal{A}_u)u \\ = \mathbf{b}_u - \mathcal{A}_u^\top \mathcal{A}_x^{-\top}(\mathbf{b}_x - W_{xx}\mathcal{A}_x^{-1}\mathbf{b}_\lambda) - W_{ux}\mathcal{A}_x^{-1}\mathbf{b}_\lambda\end{aligned}$$

- Step.2: Solve following equation to get x:

$$x = \mathcal{A}_x^{-1}(\mathbf{b}_\lambda - \mathcal{A}_u u)$$

- Step.3: Solve following equation to get λ:

$$\lambda = \mathcal{A}_x^{-\top}(\mathbf{b}_x - W_{xx}x - W_{xu}u)$$

No $LDL^\top$ of $\phi_i \rightarrow \mathcal{A}_x$ is isolated $\rightarrow \mathcal{A}_x^{-1}$ can be obtained by user defined algorithm.

Dense LU for reduced hessian (Lhs matrix in Step 1)

Numerical Results

No.Sce	n	Obj	Iters	Time(hh:mm:ss)	MPI Proc.
96	1,930,752	1.39×10^2	42	00:29:54	8
96	1,930,752	1.39×10^2	42	00:14:45	16
96	1,930,752	1.39×10^2	42	00:10:00	24
96	1,930,752	1.39×10^2	42	00:07:36	32
96	1,930,752	1.39×10^2	42	00:05:14	48
96	1,930,752	1.39×10^2	42	00:02:54	96

Table: Scalability: Reduced space (Umfpack is applied to do LU factorization and corresponding backsolve.)

No.Sce	n	Obj	Iters	Time(hh:mm:ss)	MPI Proc.
96	1,930,752	1.39×10^2	42	01:13:16	8
96	1,930,752	1.39×10^2	42	00:38:18	16
96	1,930,752	1.39×10^2	42	00:24:55	24
96	1,930,752	1.39×10^2	42	00:19:23	32
96	1,930,752	1.39×10^2	42	00:12:42	48
96	1,930,752	1.39×10^2	42	00:06:48	96

Table: Scalability: Full space

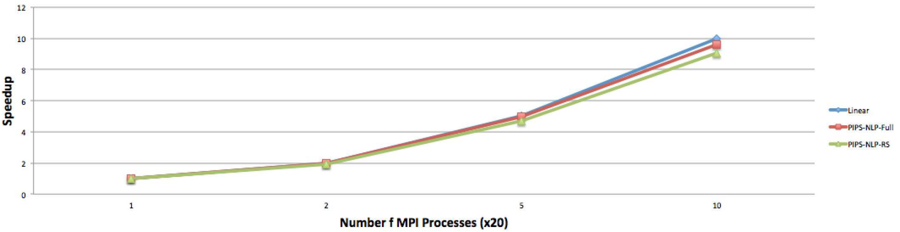
Numerical Results

No.Sce	n	Obj	Iters	Time(hh:mm:ss)	MPI Proc.
200	3,917,328	1.20×10^2	55	00:27:16	20
200	3,917,328	1.20×10^2	55	00:14:16	40
200	3,917,328	1.20×10^2	55	00:06:01	100
200	3,917,328	1.20×10^2	55	00:03:14	200

Table: Scalability: Reduced space (Umfpack)

No.Sce	n	Obj	Iters	Time(hh:mm:ss)	MPI Proc.
200	3,917,328	1.20×10^2	55	01:01:33	20
200	3,917,328	1.20×10^2	55	00:31:11	40
200	3,917,328	1.20×10^2	54	00:12:36	100
200	3,917,328	1.20×10^2	55	00:06:38	200

Table: Scalability: Full space



AMPL Input

Generate NL file from AMPL:

```
# define suffixes:
suffix pipsNLP_DecisionVar_in, IN;
suffix pipsNLP_1stStageVar_in, IN;
param idx_1stVar;
param idx_decVar;

# assign suffixes for each scenario;
for k in 1..K_All do
    let idx_1stVar := 1;
    let idx_decVar := 1;

    # define first stage variables
    for i in LINK,t in TIME: t ≤ TDEC do
        let dp[j,i,t].pipsNLP_1stStageVar_in := idx_1stVar;
        let idx_1stVar := idx_1stVar + 1;
    end for

    # define contral variables in each scenario
    for i in LINK,t in TIME: t > TDEC do
        let dp[j,i,t].pipsNLP_DecisionVar_in := idx_decVar;
        let idx_decVar := idx_decVar + 1;
    end for

    # write nl file
    write ("bpdegas_paper_Dec"& k);
end for
```

Straightforward and ‘no’ additional time for ordering derivatives (<0.1s for gas model)

Conclusions

PIPS-NLP

- Parallel NLP solver.
- Accept multiple structure, e.g: PDE constraints + network constraints.
- Support AMPL/PYOMO input.
- Other applications: parameter estimation, general stochastic optimal contral problem, robust dessign and network partitioning.

Conclusions

PIPS-NLP

- Parallel NLP solver.
- Accept multiple structure, e.g: PDE constraints + network constraints.
- Support AMPL/PYOMO input.
- Other applications: parameter estimation, general stochastic optimal contral problem, robust dessign and network partitioning.

Future Work

- Exploit multi-stage stochastic structure and multi-grid multi-level algorithm for problems with PDE and network constraints.

Conclusions

- Thank you for your attention!